

Integrating Virtual Environments with High Performance Computing

Chris Faigle, Geoffrey C. Fox, Wojtek Furmanski, Janusz Niemiec and Diglio A. Simoni

Northeast Parallel Architectures Center
111 College Place, Syracuse University
Syracuse, NY 13244-4100

submitted to the *Virtual Reality Annual International Symposium*
VRAIS '93, Sept. 18-22, Seattle, WA

January 15, 1993

Abstract

Virtual Environments will be naturally integrated with High Performance Computing since: a) advanced *Grand Challenge* type simulations will benefit from dynamic Virtual Environment front-ends, and b) advanced Virtual Environments will require simulation engines based on High Performance Computing resources. Such an integration minimally requires interoperability and standardized interfaces between both components. We extend the discussion of these requirements and illustrate how relevant design issues can be elegantly addressed by the MOVIE System, currently under development at NPAC as an infrastructure layer for dynamic interactive HPC.

Introduction The design of Virtual Environments is sometimes described as demanding the creation of entirely new computational paradigms and thus forces a fundamental rethinking of the software development process. However, we anticipate that VR will gradually permeate and build upon existing computational infrastructures. In particular, given that High Performance Computing (HPC) harmonizes with VR as both technologies represent complementary aspects of modern computing, the existence of flexible and standardized connectivity models between current and future VR front-ends and sophisticated simulation back-ends becomes a key issue. Developers of advanced *Grand Challenge* HPC simulations, developed at Supercomputer Centers and National Laboratories will continually strive to enhance their scientific visualization capabilities through the use of Virtual Environments. In turn, advanced industrial Virtual Environments will carry on with their demands for more powerful computational resources making HPC based on massively parallel and distributed processing integral components of such systems. This phenomenon already occurs in a limited sense if one regards the high-end SGI machines used in some current Virtual Environments (see e.g. [BrGY92]) as custom modest-power parallel systems. However, with the onset of a robust broadband networking infrastructure, a new form of Virtual Environment will emerge based on the concept of *Televirtuality* (see e.g. [Jaco92][Stop92] for recent reviews). Televirtual Environments are locally supported by low cost consumer VR peripherals, integrated with home/office electronics, and linked fiber-optically with supercomputing facilities which provide simulation engines. In such 21st century Global

Computing scenarios [ScAm91], various components of large applications in such diverse fields as science, economy, environment and the news will be developed by independent, geographically separated teams of individuals and so again the issue of flexible connectivity of component software models attains central importance.

In this paper we present a set of natural requirements imposed on the design of Virtual Environments which facilitates modularity, interoperability and smooth integration with HPC. We describe the Multitasking Object-oriented Visual Interactive Environment (MOVIE) System [Furm92a], currently under development at the Northeast Parallel Architectures Center (NPAC) at Syracuse University [FFHNPS92] and its relevance as a prototype model for a Televirtuality operating shell [Furm92b][FFFHN92]. We suggest some of the ways in which MOVIE addresses the proposed Virtual Environment requirements. Finally, we report on the current status of MOVIE and sketch current and planned near-term application areas.

I. Requirements for Designing Virtual Environments in HPC Figure 1 lists a minimal set of requirements for the design of HPC based Virtual Environments. These requirements are met in varying degrees by current development systems, including VEOS [BrJa92], Trix [Wals92] and VWT [PiTe92]. Most current single-user PC-based systems use a tight read-process-update loop [PiTe92] in the implementation of their Virtual Environments. Contrast this to (for example) the requirements of a multiuser system which incorporates simulations of fluid flow with pressure countour-mapped sound cues and the visualization of temperature profiles resulting in a complete Virtual Environment for exploration in the Computational Aerosciences (CAS). With the possibility of allowing *multiple users* and *multiple simulations* running concurrently, one realizes that it is more convenient to maintain the 'tick' for each simulation *separate from the VR interface device handlers*. However, the entire Virtual Environment should be time-synchronized in some globally consistent fashion across all user VR modules. The update rate of the HPC computing engine must not be tied to the update rate of the VR peripherals and vice versa. In particular, due to perceptual limitations, the frame rate must not drop below a level where simulation sickness sets in [PCC93]. The development system must allow for the simulation(s) to continue running independently, or force them to relinquish HPC resources temporarily to allow other (not necessarily related to the Virtual Environment at hand) compute-bound processes to run in order to make *efficient use of computational resources*. When confronted with using expensive HPC equipment, which may include *parallel and distributed heterogenous systems*, the development system must deal in a very careful manner with issues of real-time properties, scheduling, dynamic load-balancing, and memory management. Today's scientists use SIMD and MIMD machines side by side with workstations. They want to be able to run their simulations on the machines of their choice, and not be concerned about the interface. The development system must provide a standard user interface protocol (similar in philosophy to the X-Window system) which allows for independent customization and VR device independence. Furthermore, given the usual reluctance to change familiar means of

expressing the solution of a particular problem, the system must allow the developers of Virtual Environments to work using *familiar languages and tools*, as well as to be prepared to expand gracefully into dealing with *large amounts of data*.

- ☐ Must separate the simulation from the VR interface devices
- ☐ Must run in parallel and distributed heterogeneous environments
- ☐ Must make efficient use of resources
- ☐ Must allow multiuser participation
- ☐ Must allow multiple simulations to be run concurrently
- ☐ Must allow for development using familiar languages and tools
- ☐ Must be able to handle extremely large amounts of information

Figure 1: Requirements for Designing Virtual Environments coupled with High Performance Computing Resources

II. MOVIE and VR The MOVIE system is a network of multithreading servers which may reside on a heterogeneous network of PC's, workstations and MIMD or SIMD machines. Each server may perform its tasks independently or it may cooperate with other servers to enhance the processing efficiency of the entire ensemble. Each server has two functional layers: a kernel and an interpretive layer. The kernel is written in C and provides a base PostScript interpreter as well as scheduling, communication and memory management services. The interpretive layer is constructed via scripts written in MovieScript, a superset of PostScript. Some design concepts are inherited from the Sun NeWS server design [GRA89]. A network of MOVIE servers offers an abstraction of a homogeneous virtual machine on top of a heterogeneous distributed hardware. For example, MOVIE has built in matrix algebra. A MOVIE server for a SIMD machine will execute a matrix multiplication much differently than one running on a multicomputer or a workstation. This, however, is completely transparent to a thread which invokes this operation. A scheduler controls the interpreted threads, while the communication manager ensures communication between threads within one MOVIE server as well as between threads belonging to other MOVIE servers. The MovieScript interpreter translates a thread's code into C-coded kernel functions. Because the implementation makes few assumptions about the underlying hardware and software, MOVIE is highly portable. The efficient multithreading, scheduling and hierarchical communication schemes of MOVIE fulfill many of the requirements for runtime support specified in the recent DARPA Draft [DRWG92].

Multithreading Threads are scheduled in a preemptive fashion according to their priority level and assigned to one of two types of multilevel queues: *real-time* queues or *computational* queues. Real-time threads with higher priorities always preempt lower priority threads. Because all scheduling functions are accessible from the interpretive level, users can easily build a custom scheduler in MovieScript and replace the default one at run time. In order to keep up the frame

rate, for example, threads that handle VR peripherals may preempt threads that handle simulation-related compute-bound operations. Thus, in a multiple-simulation Virtual Environment, the scheduling priorities can be adjusted to ensure that the user interface is updated at a comfortable level. This may come at the expense of simulation processing in a single server environment, yet these detrimental effects decrease in a multiserver MOVIE ensemble. Threads can become *blocked* so multiple simulations can therefore run on the same MOVIE server providing a faster response than if they were compiled independently and ran via machine time-sharing.

Communication When cooperating threads need to synchronize their activities, they may use semaphores or direct-addressed synchronous/asynchronous message passing (the lowest communication method). If no access to shared objects is required, a send/wait operation is a proper way to synchronize. The order of thread execution may be established by using special kinds of messages called *tickets*. The synchronization requests are carried out by the scheduler, which suspends a thread when an operation is unable to continue or makes a thread ready. Since all threads are interpreted, they can migrate from one MOVIE server to another in order to perform efficient dynamic load-balancing as new Virtual Environment simulations are started, or the computational requirements of existing ones change. Given that message passing introduces the lowest overhead, threads which perform time-critical operations or are bound to a particular node of the MOVIE network communicate using this method. When an application such as a multiple-user Virtual Environment demands many operations on shared, cooperative objects, a Linda-like communication layer [Gele85] is more appropriate. This model is based on a global tuple space which accepts newly created tuples and allows inspection by all threads. Every tuple contains a MovieScript object, and in particular a thread's code may be a content of such tuple.

III. Current Status of MOVIE The MOVIE System is based on a custom made high performance skeleton PostScript interpreter with graphics interfaces to NeWS and DPS servers and it currently includes the following computational domains, structured as MovieScript extensions of PostScript: preemptive multithreading, multilevel queue based scheduling with real-time support, uniform communication protocol for networking (distributed computing) and message passing (multicomputing), support for C++ style interpretive object-oriented programming with multiple inheritance and dynamic binding, index-free matrix algebra of the APL/Fortran90 type, interfaces to Xlib and X based GUI toolkits such as Motif, a 3D graphics model, portable between PHIGS, PEX and GL drivers, and an interface to dataflow based scientific visualization systems such as AVS. These components will be available in an integrated and documented form within the first internal release of MOVIE 1.0, scheduled for May '93. We are now developing the High Performance Fortran Interpreter (HPFI) [FoFu92][FFFHN92] on top of MOVIE, to be demonstrated at Supercomputing '93. Preliminary MOVIE/HPFI demos will be available in the summer/fall '93. HPFI builds on top of NPAC's active involvement in national HPC programs, aimed at developing HPF language protocol [FHKKKTW90] and prototype HPF compiler [BCFHR93], and it will extend NPAC's expertise in domains of parallel processing such as science [FJLOS88]

and industry [Fox92] applications towards advanced dynamic simulations, such as pursued by SIMNET and follow-on programs [Zyda92]. Within the new VR lab, planned for '93 at NPAC, we will also start exploring detailed connectivities between MOVIE and current generation of VE development environments such as VEOS, TRIX and VWT.

As a near term application area, we intend to explore the educational potential of Televirtuality [Gore91][NRC92]. Selected *Grand Challenge* applications will be installed at NPAC's HPC facilities which include CM-5, DECmpp, Intel Touchstone and nCUBE2 running parallel Oracle, restructured in terms of MOVIE/HPFI tools into dynamic object-oriented formats required by VR and to suitable levels of K-12 education, and delivered to local area schools. A fiber-optic OC3 line is currently being developed by NYTel/NYNEX in Central New York within the Upstate Corridor of NYNET which will connect the HPC facilities at Syracuse University, Cornell University and Rome Laboratories.

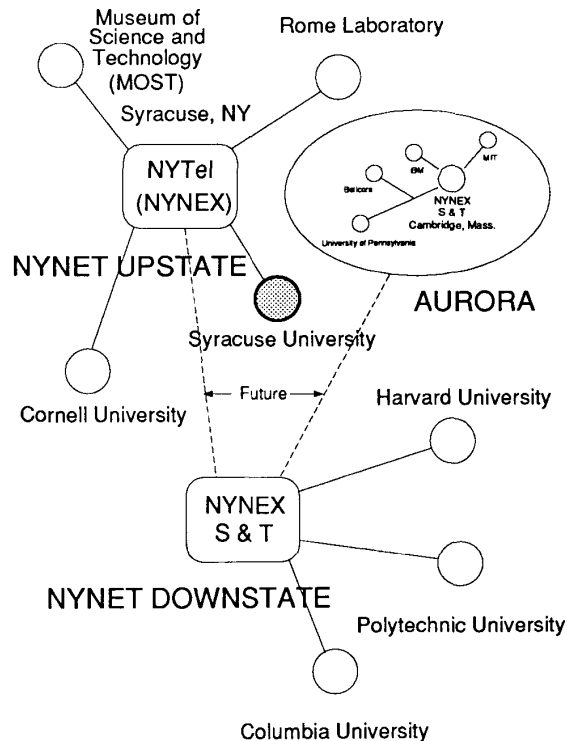


Figure 2: Current and future connectivities within NYNET, a high speed fiber-optic communication network linking multiple computing, communication and research facilities in New York State. NYNET is composed of Upstate and Downstate corridors. The Upstate corridor will integrate advanced computer research at Cornell University's Supercomputing and Syracuse University's Parallel Processing Centers, military-based research at Rome Laboratories, and communications research at NYNEX. That connectivity will extend to such facilities as Columbia, Harvard and polytechnic Universities. Future goals look towards eventual links to the Aurora Testbed and other National Research and Education Network (NREN) sponsored networks.

This network will be linked in the next stage with the Downstate Corridor and with the Aurora testbed within the National Research and Education Network (NREN). We expect early access to new consumer VR products, developed by Abrams/Gentile Entertainment (AGE) and we plan to explore concepts of Televirtuality by installing these devices in selected schools along the NYNET line and linking them via video dialtone lines to virtualized *Grand Challenge* simulations running at NPAC.

As an example of local VR peripheral research work, we mention our ongoing efforts [Simo93] on implementing a static gesture recognition Content Addressable Memory (CAM) using HPF. As the cost of operating the CAM increases with the number of gestures and the number of sensor signals, there is an obvious need for computational resources allocated exclusively to the recognition task. Another ongoing research effort involves the exploration of HPC resources applied to the type of VR interface pioneered by Krueger [Krue91]. This method reduces to using image processing techniques coupled to an image interpretation system. Most of the tasks involved can be efficiently implemented in HPF.

Conclusion High Performance Computing has very specific needs regarding the development of Virtual Environments. Most scientific simulations of interest push beyond the capabilities provided by even today's Gigaflop computers. Providing VR interfaces to these simulations, or conversely using these simulations in the construction of a standard Virtual World presents us with special challenges. These include Operating System, architecture, networking, load-balancing, extremely large data set and simulation issues. The MOVIE system, being a collection of MOVIE servers which run on any combination of heterogeneous workstations and/or parallel machines, reduces these resources to a homogeneous architecture, and provides low and high level services, including multithreading, scheduling, networking, language interpreters, and high level object abstraction. Comparing the services provided by the MOVIE servers, to the requirements for HPC based Virtual Environments, we find that the MOVIE system is a close match. MOVIE thus fills a large gap when coupling Virtual Environments with HPC resources, using experience from both the VR and the HPC communities.

References

- [BrGY92] Bryson, S. and Gerald-Yamaski, M., "The Distributed Virtual Windtunnel", RNR Technical Report RNR-92-0130, NASA Ames Research Center, Moffett Field, CA, 1992.
- [BrJa92] Bricken, W. and Jacobson, L., "Virtual Environment Operating System", in *AI Expert: Virtual Reality Special Report*, July 1992
- [BCFHR93] Bozkus, Z., Choudhary, A., Fox, G.C., Haupt, T., Ranka, S., "Distributions in F90D" and "Forall in F90D", SCCS TR93 (in preparation)
- [DRWG92] "DARPA Runtime Workshop Group: A Draft Outline", Internal NPAC Report, 1992.
- [Fox92] Fox, G.C., "Parallel Computing in Industry - An Initial survey", Syracuse Center for Computational Science, Technical Report, 1992.

- [FoFu92] Fox, G.C. and Furmanski, W., "Towards Interpreted Run-Time Fortran90D Environment", Syracuse Center for Computational Science, Internal Report, 1992.
- [Furm92a] Furmanski, W., "MOVIE – Multitasking Object-oriented Visual Interactive Environment", in Fox, G.C., Messina, P. and Williams, R., editors, *Parallel Computing Works*, Morgan and Kaufman, 1992 (in preparation).
- [Furm92b] Furmanski, W., "Supercomputing and Virtual Reality", paper presented at the Meckler Conference *Virtual Reality '92*, San Jose, CA, Sept 23–25, 1992.
- [FFHNPS92] Faigle, C., Furmanski, W., Haupt, T., Niemiec, J., Podgorny, M. and Simoni, D., "MOVIE Model for Open Systems based High Performance Distributed Computing", in *Proceedings of the First International Symposium on High Performance Distributed Computing, HPDC-1 Sept 9–10, 1992, Syracuse, NY*, IEEE Computer Society Press.
- [FFFHN92] Faigle, C., Fox, G.C., Furmanski, W., Haupt and T., Niemiec, "Software Integration towards Global Computing", paper submitted to the 7th ACM *International Conference on Supercomputing*, Tokyo, Japan, July 19–23, 1993.
- [FHKKKTW90] Fox, G.C., Hiranadani, S., Kennedy, K., Koelbel, Ch., Kremer, U., Chau-Wen Tseng and Min-You Wu, "Fortran D Language Specification", SCCS/42C, 1990.
- [FJLOSW88] Fox, G.C., Johnson, M.A., Lyzenga, G.A., Otto, S.W., Salmon, J.K. and Walker, D.W., *Solving Problems on Concurrent Processors*, Prentice-Hall, 1988.
- [Gele85] Gelernter, D., "Generative communication in Linda", *ACM Trans. on Programming Languages and Systems*, 7(1):80–112, 1985.
- [Gore91] Gore, A., "Infrastructure for the Global Village", in *Scientific American*, September 1991.
- [GRA89] Gosling, J., Rosenthal, D.S.H., Arden, M., *The NeWS Book*, Springer Verlag, 1989.
- [Jaco92] Jacobson, R., "Televirtuality: Status of Projects Around the World", talk presented at the EFDPMAC Conference on Virtual Reality, Washington, DC, June 1–2, 1992.
- [Krue91] Krueger, M.W., *Artificial Reality II*, Addison-Wesley, 1991.
- [NRC92] National Research Council, Memorandum on "NAS/NRC Planning Meeting on Virtual Reality", January 1992.
- [PiTe92] Pimentel, K. and Teixeira, K., *Virtual Reality: Trough the new looking glass*, Widcrest Books, McGraw-Hill, Inc., 1992.
- [PCC93] Pausch, R., Crea, T. and Conway, M., "A Literature Survey for Virtual Environments: Military Flight Simulation Visual Systems and Simulator Sickness", to appear in *Presence*, MIT Press.
- [ScAm91] "Communications, Computers and Networks: How to Work, Play and Thrive in Cyberspace", cover story in *Scientific American*, September 1991.
- [Simo93] Simoni, D.A., "Polynomial Network CAMs for Static Gesture Recognition in Virtual Environments", (in preparation)
- [Stop92] Stoppi, J.U.L., "Televirtual Environments: Concept, Applications, and Implications", in Sandra K. Helsel, ed., *Virtual Reality Report*, Meckler, Sept 1992.
- [Wals92] Walser, R., "A De-Facto Anti-Standard for Cyberspace", talk presented at the Meckler Conference *Virtual Reality '92*, San Jose, CA, Sept 23–25, 1992.
- [Zyda92] Zyda, M., "The Software Required For the Computer Generation of Virtual Environments", talk presented at the 3rd EFDPMAC Conference on Virtual Reality, Washington, DC, Dec 1–2, 1992.